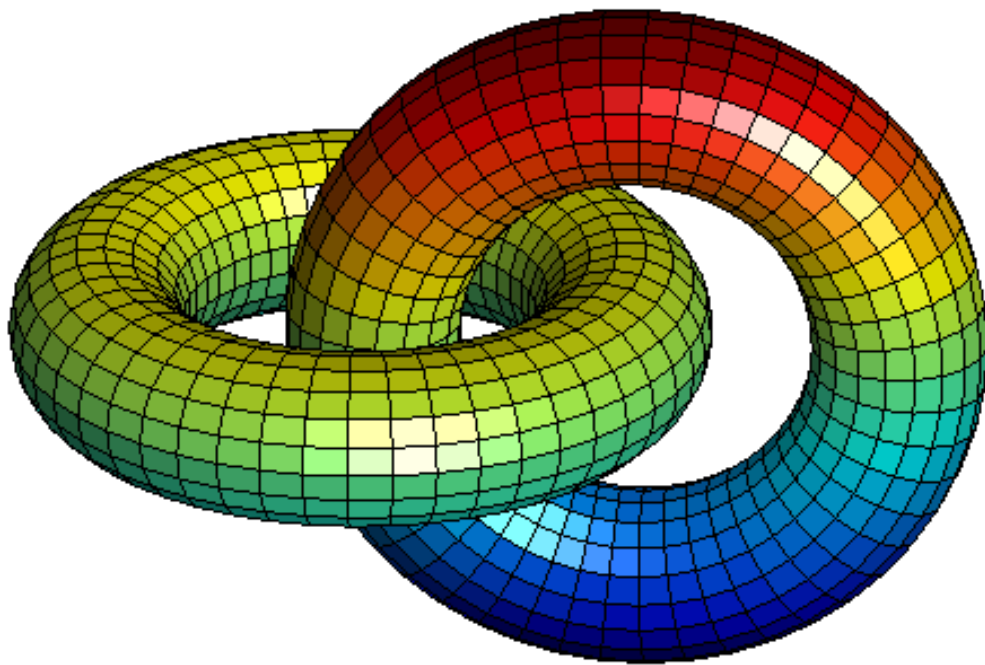


# Práctica 1

## Visualización en MATLAB



**Héctor Garbisu Arocha**

Curso 2015/16

Métodos Numéricos para la Computación

Grado en Ingeniería Informática

Escuela de Ingeniería Informática

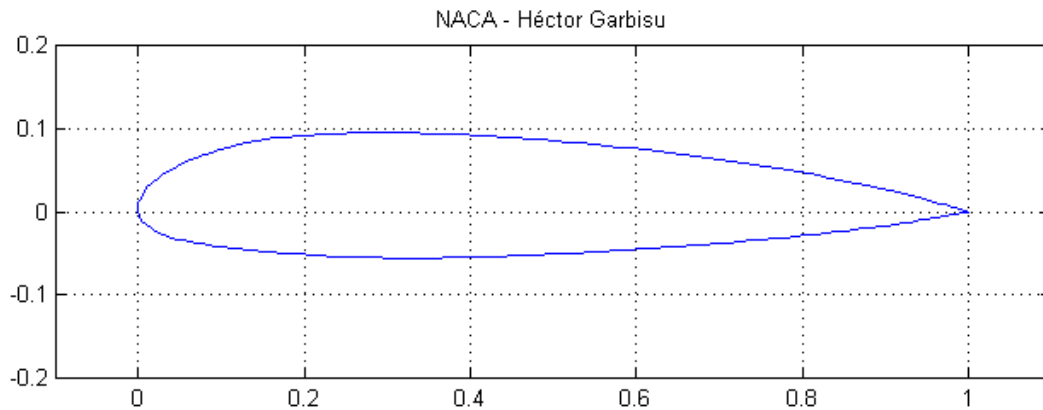
Universidad de Las Palmas de Gran Canaria

# Índice

1. Airfoil NACA .....	pág. 3
2. Airfoil 644XX .....	pág. 4
3. Esfera y Toro .....	pág. 4
4. Imagen plana 1 .....	pág. 5
5. Imagen proyectada en plano .....	pág. 6
6. Imagen proyectada en superficie curva.....	pág. 7
7. The One Ring .....	pág. 7
8. The One Ring 2 .....	pág. 8
9. Código .....	pág. 9
10. Bibliografía .....	pág. 14

## 1. Airfoil NACA

Este ejercicio no ha tenido complicación alguna. Los pasos a seguir en el gui3n de la pr3ctica son claros y se han podido hacer sin incidencias.



Para guardar la imagen he usado la conversi3n que hace Matlab en lugar de hacer captura de pantalla. El motivo es que as3 se transforma autom3ticamente el fondo en color blanco, ideal para la memoria.

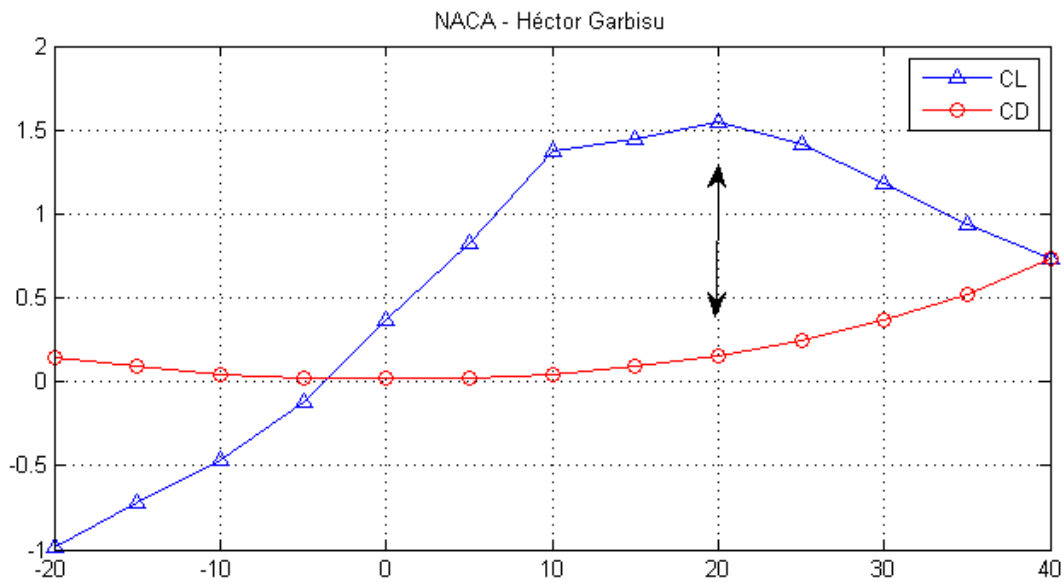
El c3digo est3 al final de la memoria.

[Enlace al c3digo](#)

## 2. Airfoil 644XX

En esta pod3amos abusar del hold para hacer dos plots consecutivos en la misma gr3fica o hacer el plot con las dos funciones a la vez. Eleg3 la segunda opci3n.

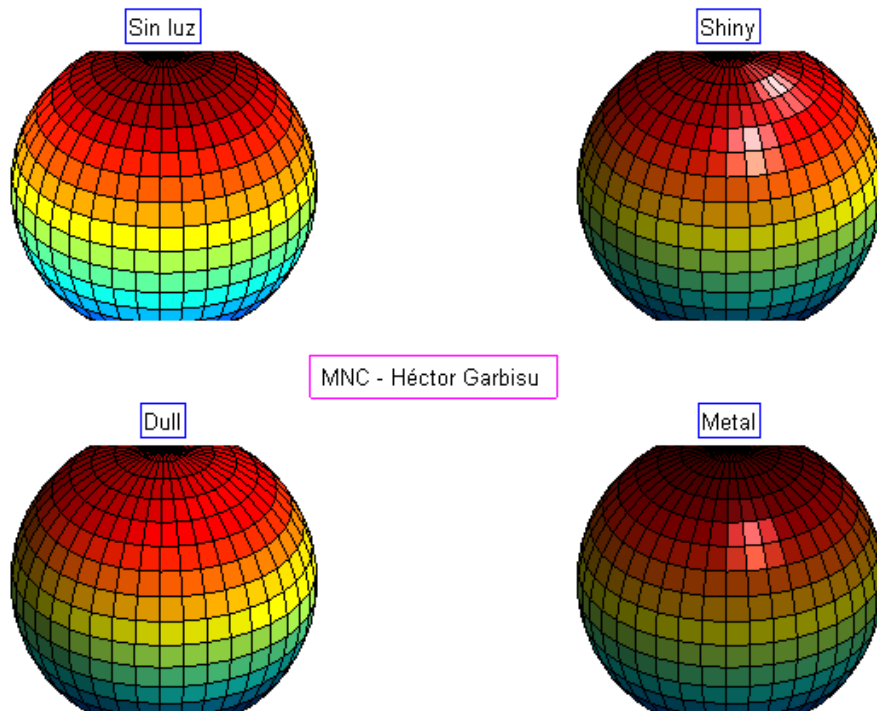
En el postproceso inclu3 una flecha doble para se3alar una separaci3n entre las dos secuencias de puntos.



El código está al final de la memoria.  
[Enlace al código](#)

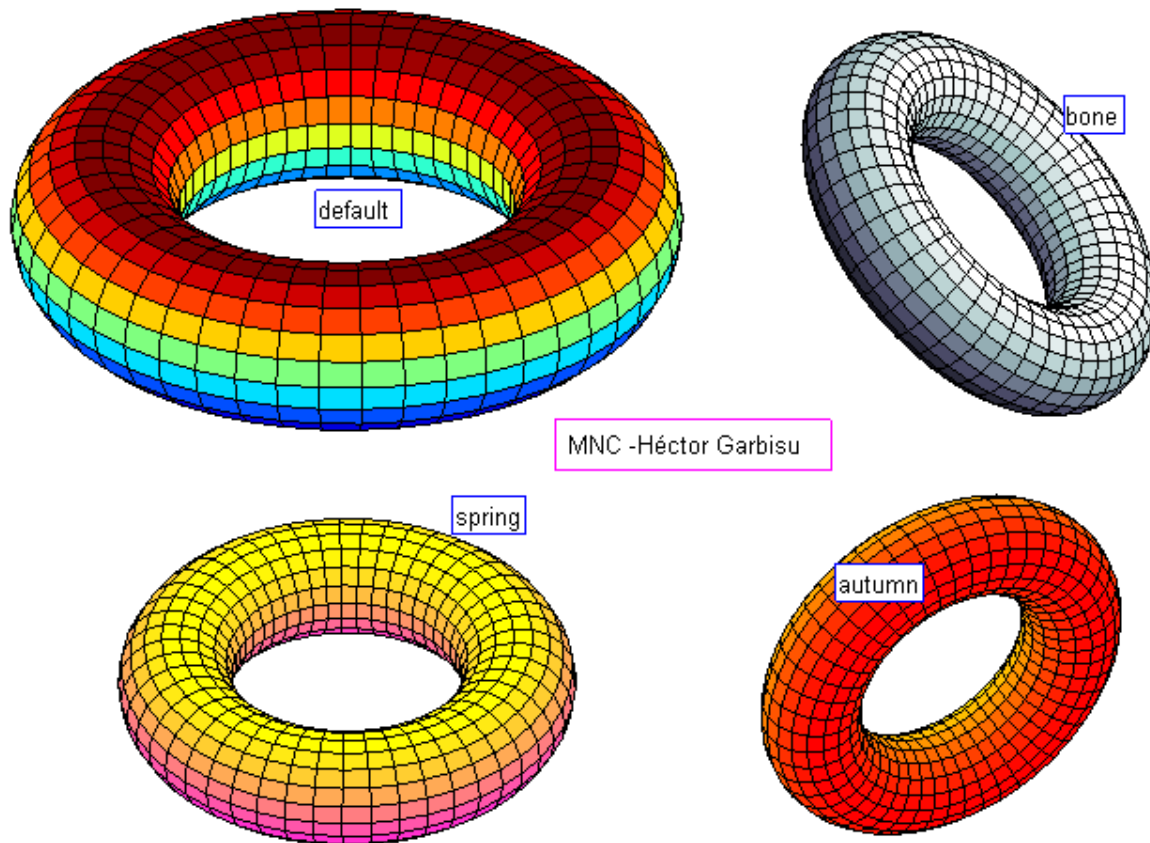
### 3. Esfera y Toro

Utilicé la visualización de la esfera para mostrar varios materiales y la del toro para mostrar otras características.



Haciendo cuatro subplots se genera una figura en la que caben cuatro imágenes como esa. En postproceso he añadido mi nombre, ya que no sabía una forma fácil de poner un título en lo alto de varios subplot.

El código está al final de la memoria. [Enlace al código](#)



Hay muchos mapas de colores predefinidos. Éstos son cuatro de ellos. También he dispuesto diferentes zooms y rotaciones.

Este apartado ha sido más complicado porque el comando `colormap` afecta al gráfico entero, no a cada subplot. He aplicado una función llamada `freezeColors` (John Iversen, 2005-10) que encontré por internet.

[Enlace al código.](#)

## 4. Visualización de una Imagen

La primera forma de visualizar una imagen es muy simple: `image(imagen)`, donde `imagen` es `imread(ruta_imagen)`

Teddy - Héctor Garbisu



[Enlace al código.](#)

## 5. Imagen proyectada en plano

A diferencia del ejercicio anterior, aquí tiene sentido manipular la imagen. Por eso se incluye cameratoolbar.

Al rotar la imagen, el título sale fuera de la vista. El texto con mi nombre lo puse con un cuadro de texto en postproceso otra vez.

MNC - Héctor Garbisu



[Enlace al código.](#)

## 6. Imagen proyectada en superficie curva

Al igual que en el caso anterior, se usa el primer canal de la imagen (como es en blanco y negro da igual cual coger) y se utiliza como patrón de coloreado, sustituyendo a la altura, que es el valor por defecto. La mayor diferencia es que el vector Z en vez de ser todo ceros, ahora está en función de X e Y, formando un paraboloide.

MNC - Héctor Garbisu



[Enlace al código.](#)

## 7. The One Ring

La definición de un colormap es tan simple como disponer 64 valores RGB seguidos, o eso deduzco al ver que todos los mapas de colores por defecto tienen esa cantidad. Aplicar el mapa de colores oro pepita como indica el guión de la práctica genera el siguiente anillo.

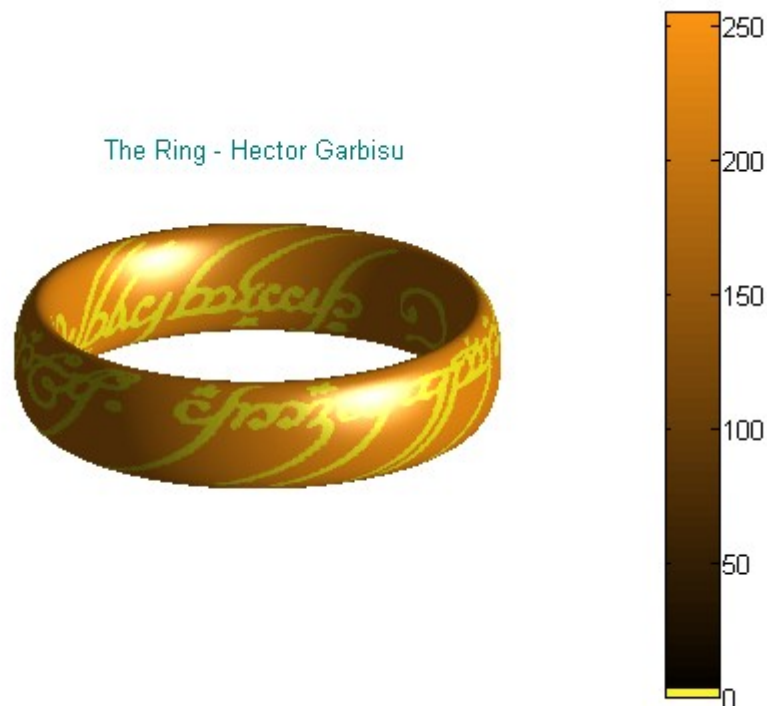


[Enlace al código.](#)

## 8. Mejoras The One Ring

Para enrojecer un poco el mapa de colores he reducido un 15% la intensidad de los canales verde y azul (multiplicando por 0.85) y dejando intacto el canal rojo. El primer valor de la tabla, con el que se pintarán los símbolos del anillo, se conserva el color áureo original.





[Enlace al código.](#)

## 9. Código

### 9.1 Airfoil NACA

```
clear all;
clc;
figure;
hold on;
grid on;
axis equal;
axis([-0.1 1.1 -0.2 0.2]);
whitebg('white');
title('NACA - Héctor Garbisu')
[x,y] = NACA4(2215);
plot(x,y,'b-');
box on
hold off;
```

### 9.2 Airfoil 644XX

```
clear all;
clc;
figure;
hold on;
grid on;
```

```
axis([-20 40 -1 2]);
title('NACA - Héctor Garbisu')
airfoil = AirfoilNACA644X(20.);
plot(airfoil.alpha,airfoil.CL,'b^-',airfoil.alpha,airfoil.CD,'ro-');
legend('CL','CD')
box on
hold off;
```

## 9.3.1 Esferas

```
clear all;
clc;
%% Qué es una esfera
R = 1;
Nt = 20;
Np = 40;
theta = linspace(0,pi,Nt);
phi = linspace(0,2*pi,Np);
[Theta,Phi] = meshgrid(theta,phi);
X = R*sin(Theta).*cos(Phi);
Y = R*sin(Theta).*sin(Phi);
Z = R*cos(Theta);
%% primera esfera
subplot(2,2,1)
surf(X,Y,Z);
axis equal;
axis([-1 1 -1 1 -0.2 0.2])
axis off;
title('Sin luz','Background','White','EdgeColor','Blue')
%% segunda esfera
subplot(2,2,2)
surf(X,Y,Z);
axis equal;
axis off;
axis([-1 1 -1 1 -0.2 0.2])
light('Position',[0 -1 1])
light
material shiny
title('Shiny','Background','White','EdgeColor','Blue')
%% tercera esfera
subplot(2,2,3)
surf(X,Y,Z);
axis equal;
axis off;
axis([-1 1 -1 1 -0.2 0.2])
light('Position',[0 -1 1])
light
material dull
title('Dull','Background','White','EdgeColor','Blue')
%% cuarta esfera
subplot(2,2,4)
surf(X,Y,Z);
axis equal;
axis off;
axis([-1 1 -1 1 -0.2 0.2])
light('Position',[0 -1 1])
material metal
title('Metal','Background','White','EdgeColor','Blue')
%%
set(gcf,'color','black');
```

## 9.3.2 Toros

```
clear all;
clc;
R1 = 3;
R2 = 1;
Nt = 20;
Np = 50;
theta = linspace(0,2*pi,Nt);
phi = linspace(0,2*pi,Np);
[Theta,Phi] = meshgrid(theta,phi);
X = (R1 + R2*cos(Theta)).*cos(Phi);
Y = (R1 + R2*cos(Theta)).*sin(Phi);
Z = R2*sin(Theta);
cameratoolbar
%% primer toro
subplot(2,2,1)
surf(X,Y,Z);
axis equal;
axis off;
colormap default
freezeColors
title('default','Background','White','EdgeColor','Blue')
%% segundo toro
subplot(2,2,2)
surf(X,Y,Z);
axis equal;
axis off;
colormap bone
freezeColors
title('bone','Background','White','EdgeColor','Blue')
%% tercer toro
subplot(2,2,3)
surf(X,Y,Z);
axis equal;
axis off;
colormap spring
freezeColors
title('spring','Background','White','EdgeColor','Blue')
%% cuarto toro
subplot(2,2,4)
surf(X,Y,Z);
axis equal;
axis off;
colormap autumn
title('autumn','Background','White','EdgeColor','Blue')
%%
set(gcf,'color','black');
```

## 9.4 Visualización de una Imagen

```
clear
imagen = imread('teddy.jpg');
image(imagen)
axis equal;
axis off;
whitebg('black');
title('Teddy - Héctor Garbisu');
```

```
hold off;
```

## 9.5 Imagen proyectada en plano

```
imagen = imread('charlize.jpg');
Nf = size(imagen,1);
Nc = size(imagen,2);
x1 = linspace(0,1,Nc);
y1 = linspace(0,1,Nf)*Nf/Nc;
[X,Y] = meshgrid(x1,y1);
Z = zeros(Nf,Nc);
C = double(imagen(:,:,1));
hold on;
surf(X,Y,Z,C);

shading interp;
colormap gray;
axis equal;
axis off;
whitebg('black');
title('Charlize - Héctor Garbisu');
hold off;
cameratoolbar;
```

## 9.6 Imagen proyectada en superficie curva

```
clear
figure('Color','Black')
imagen = imread('charlize.jpg');
Nf = size(imagen,1);
Nc = size(imagen,2);
x1 = linspace(0,1,Nc);
y1 = linspace(0,1,Nf)*Nf/Nc;
[X,Y] = meshgrid(x1,y1);
Z = 0.7*(X.^2+Y.^2);
C = double(imagen(:,:,1));
hold on;
surf(X,Y,Z,C);
% shading interp;
colormap copper;
shading interp
axis equal;
axis off;
whitebg('black');
title('Charliiize');
hold off;
cameratoolbar;
```

## 9.7 Anillo

```
clear
tabla = ones(64,3);
base = [250 174 24]./255;
for a=1:64
    tabla(a,:) = (base*a/64);
end
```

```

oro pepita=tabla;
colormap(oro pepita);

imagen = imread('tolkien.gif');
Nf = size(imagen,1);
Nc = size(imagen,2);
C = double(imagen(:,:,1));
R2 = 1;
R1 = 3*Nc*R2/Nf;
theta = linspace(0,2*pi,Nf);
phi = linspace(0,2*pi,Nc);
[Theta,Phi] = meshgrid(theta,phi);
X=(R1+R2.*sin(Theta)).*cos(Phi);
Y=(R1+R2.*sin(Theta)).*sin(Phi);
Z=2.5*R2.*cos(Theta);
material shiny
surf(X,Y,Z,C');
title('\color{rgb}{0 0.5 0.5}The Ring - Hector Garbisu')
shading interp;
camlight
axis equal;
axis off;
colorbar;
cameratoolbar;

```

## 9.7 Anillo con mejoras

```

clear
tabla = ones(64,3);
base = [250 174 24]./255;
for a=2:64
    tabla(a,1) = (base(1)*a/64);
    tabla(a,2:3) = (base(2:3)*a/64)*0.85;
end
tabla(1,:) = [255 255 0]./255;

oro pepita2=tabla;
colormap(oro pepita2);

imagen = imread('tolkien.gif');
Nf = size(imagen,1);
Nc = size(imagen,2);
C = double(imagen(:,:,1));
R2 = 1;
R1 = 3*Nc*R2/Nf;
theta = linspace(0,2*pi,Nf);
phi = linspace(0,2*pi,Nc);
[Theta,Phi] = meshgrid(theta,phi);
X=(R1+R2.*sin(Theta)).*cos(Phi);
Y=(R1+R2.*sin(Theta)).*sin(Phi);
Z=2.5*R2.*cos(Theta);
material shiny
surf(X,Y,Z,C');
title('\color{rgb}{0 0.5 0.5}The Ring - Hector Garbisu')
shading interp;
camlight
axis equal;
axis off;
colorbar;
cameratoolbar;

```

## 10. Bibliografía

A parte del material facilitado por la asignatura, se ha usado código presente aquí, como parte de un esfuerzo opcional de representar varias imágenes a la vez.

<http://www.mathworks.com/matlabcentral/fileexchange/45923-qav-pet--quantitative-analysis-and-visualization-of-pet-images/content/freezeColors.m>